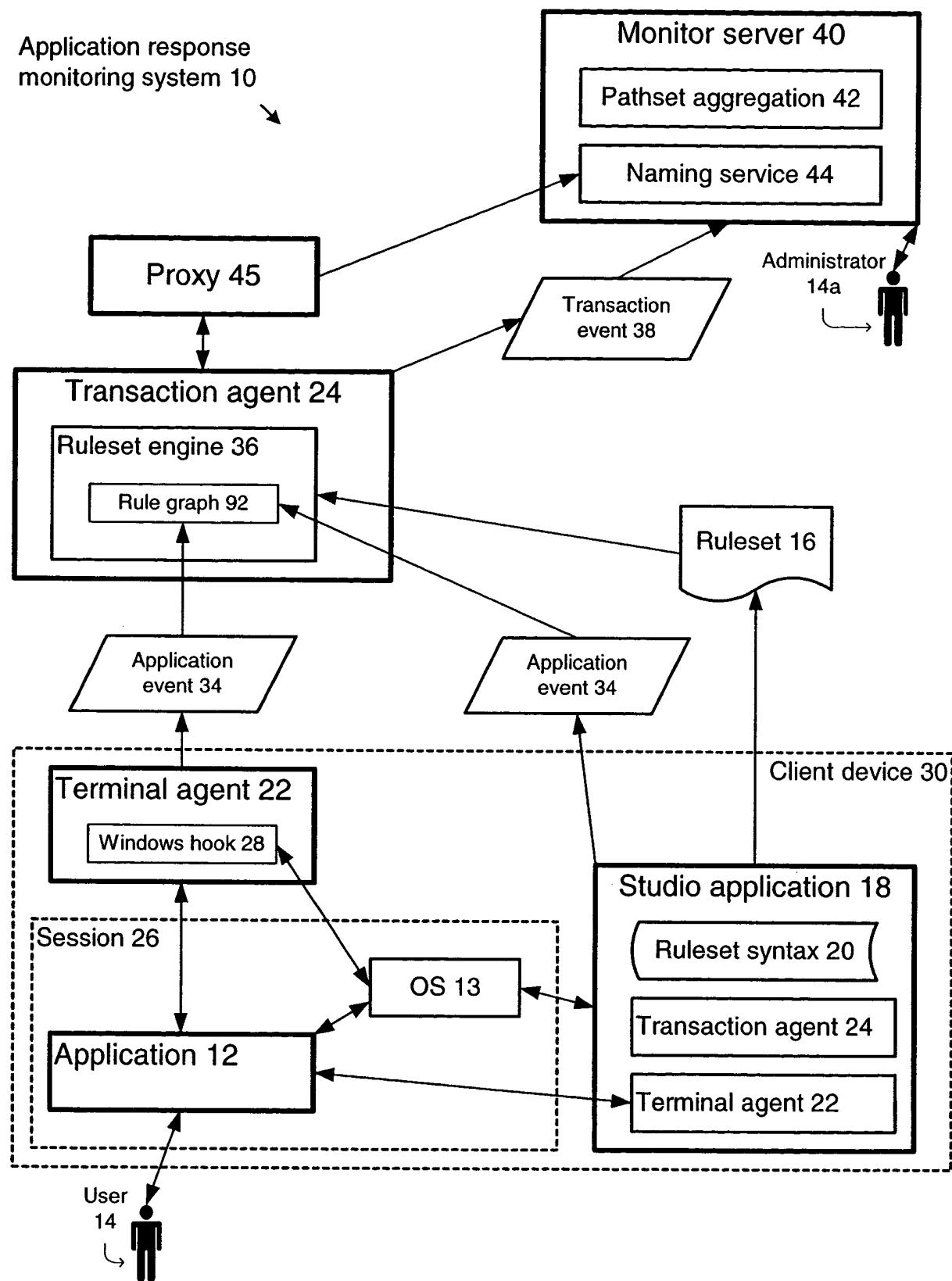
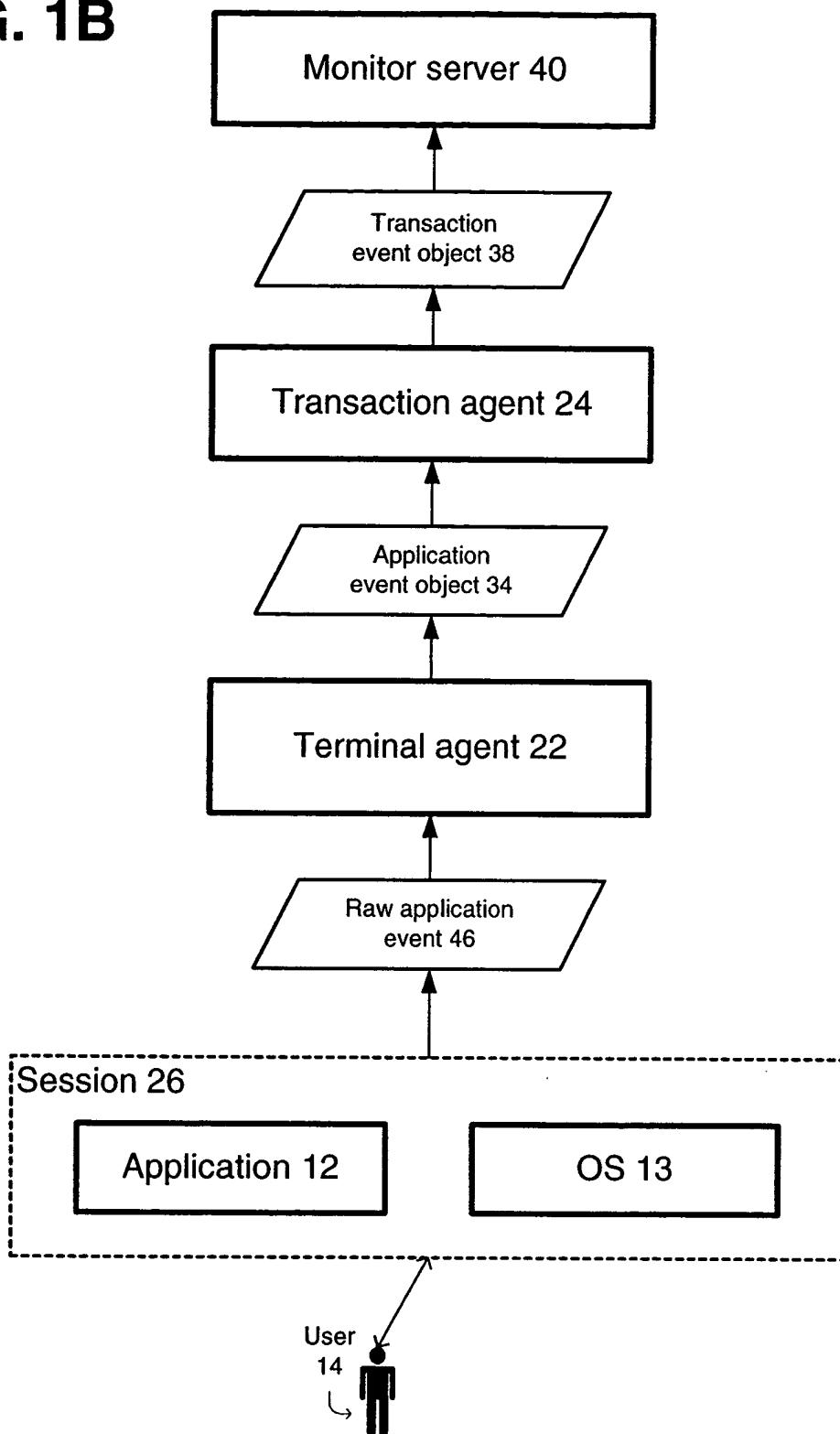


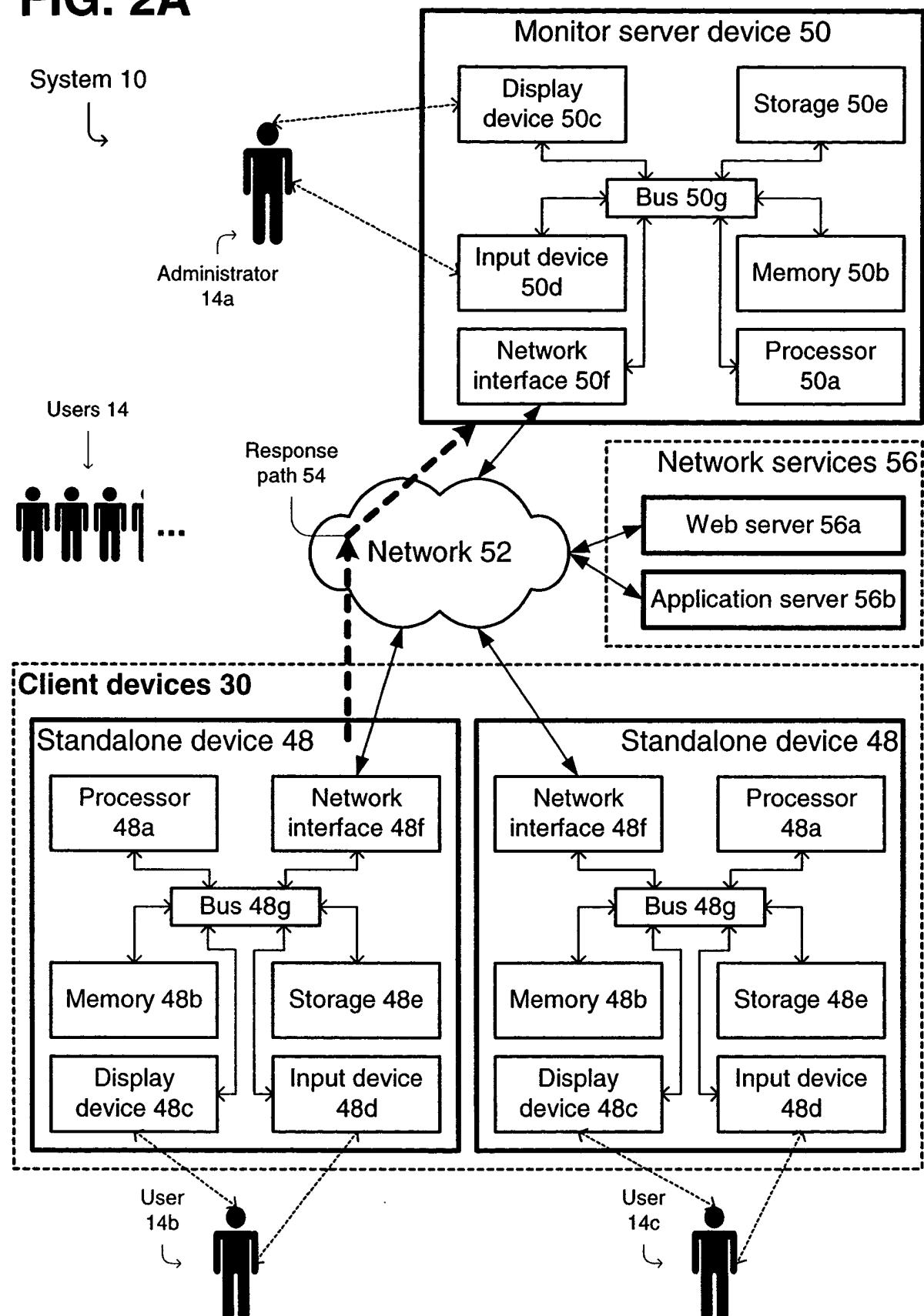
# FIG. 1A



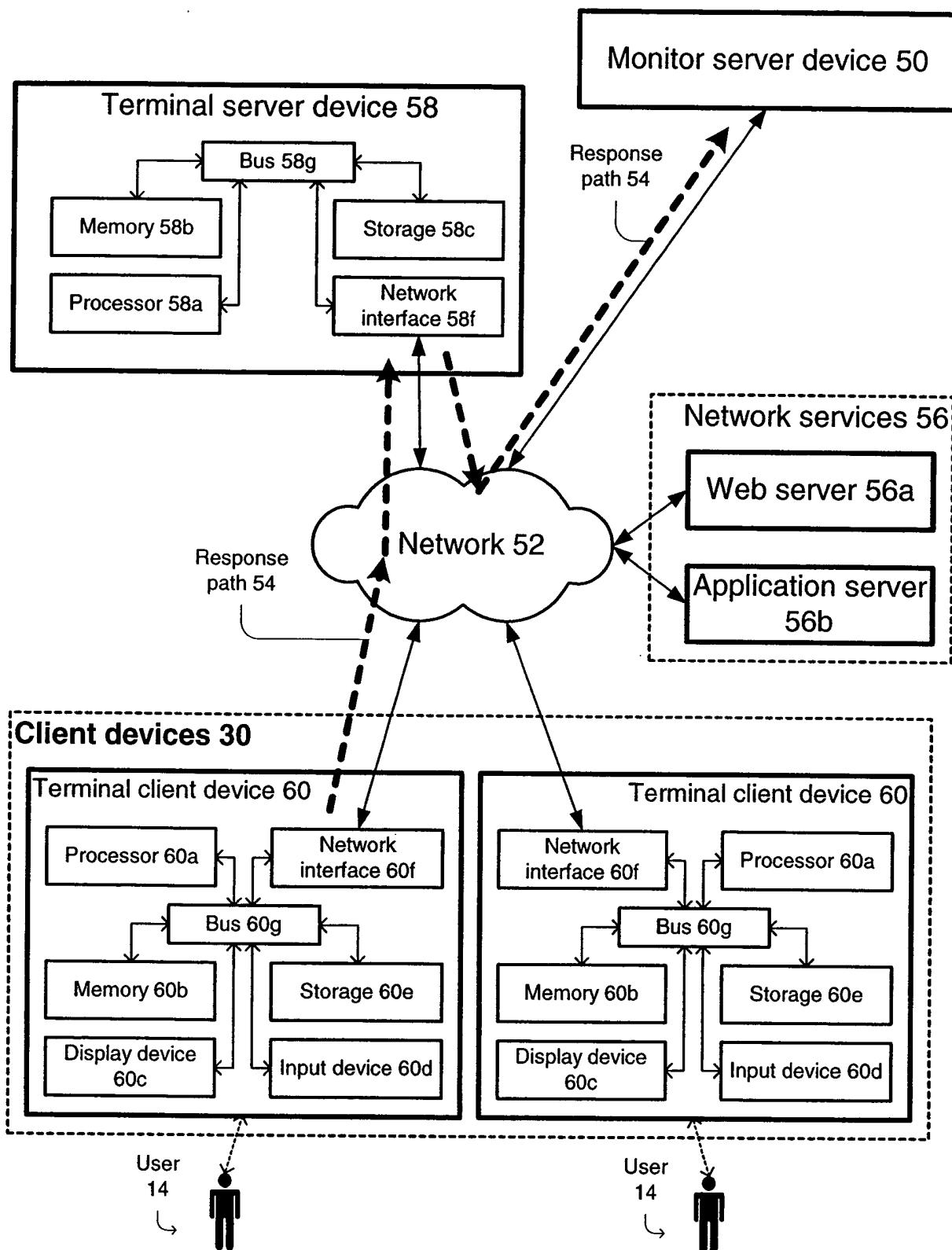
**FIG. 1B**



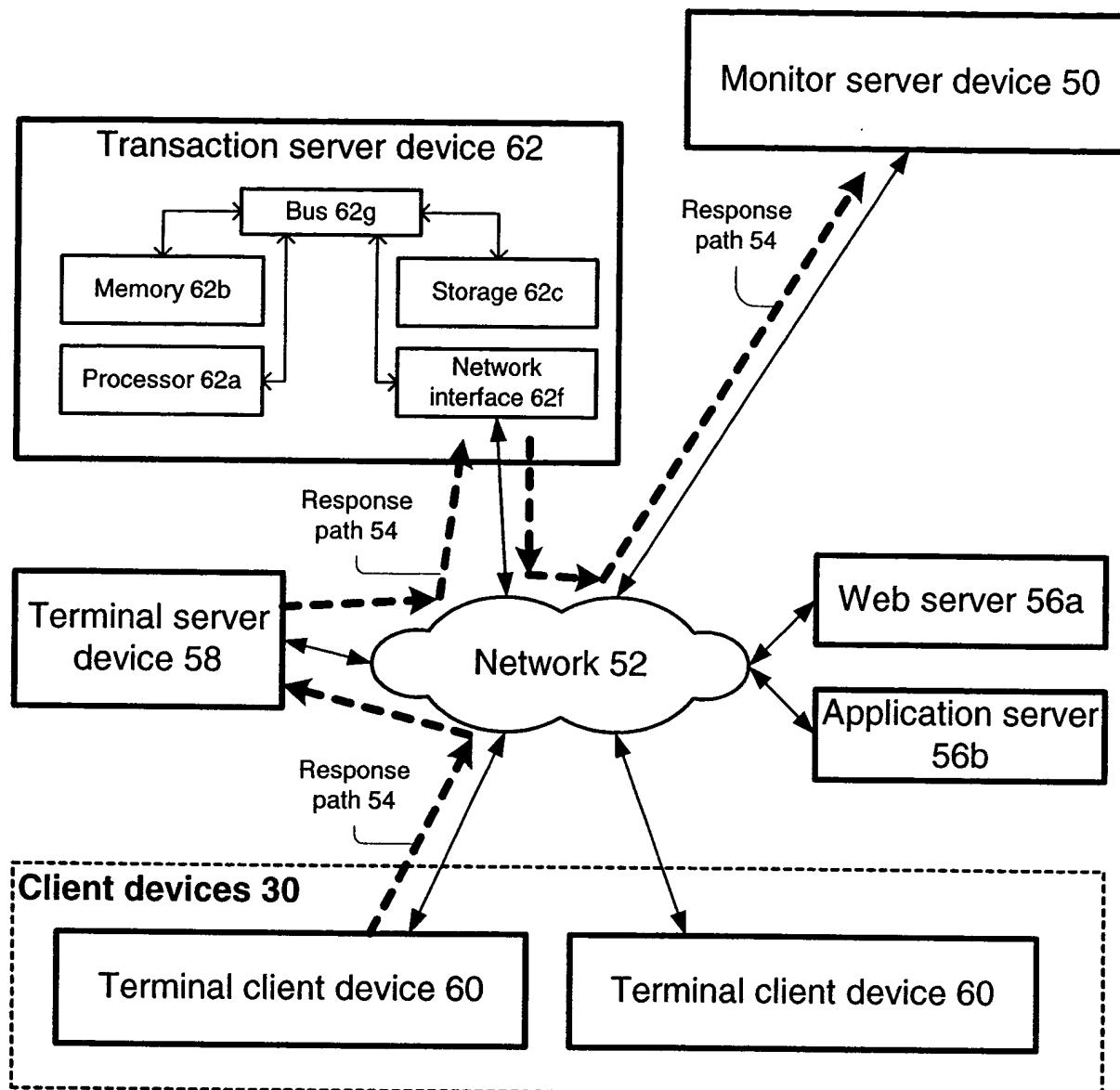
# FIG. 2A



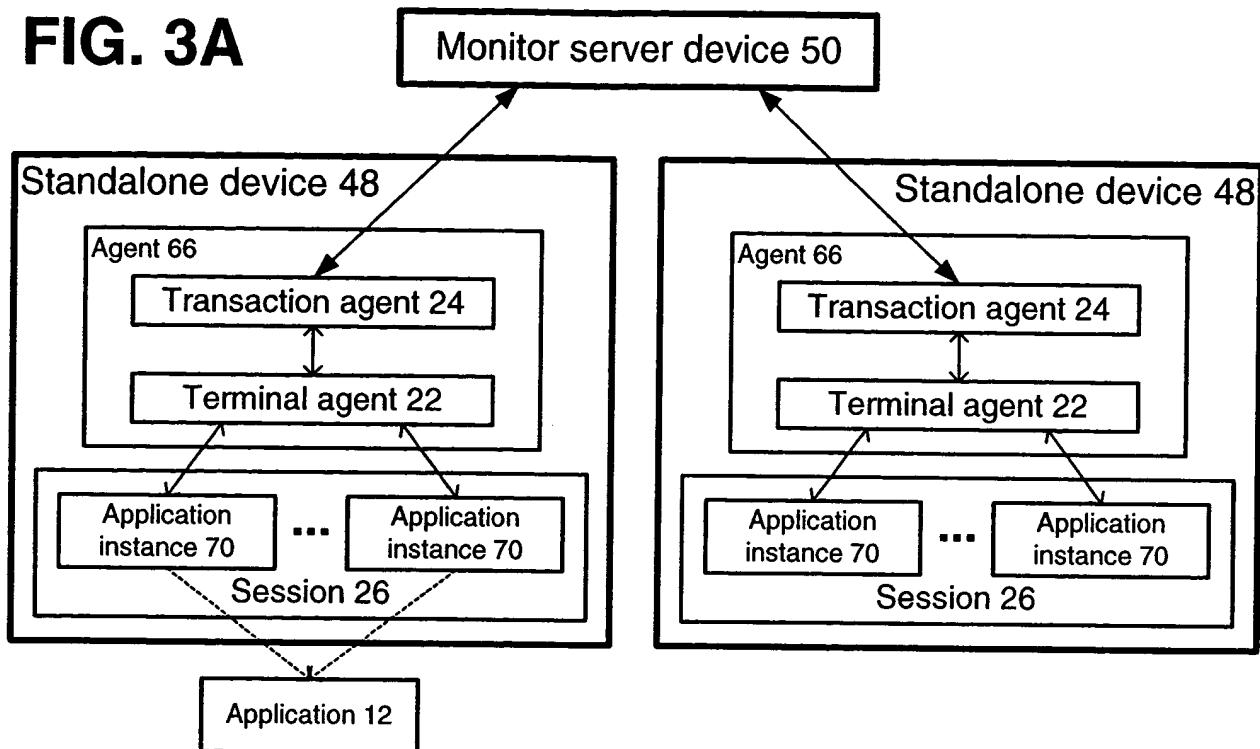
# FIG. 2B



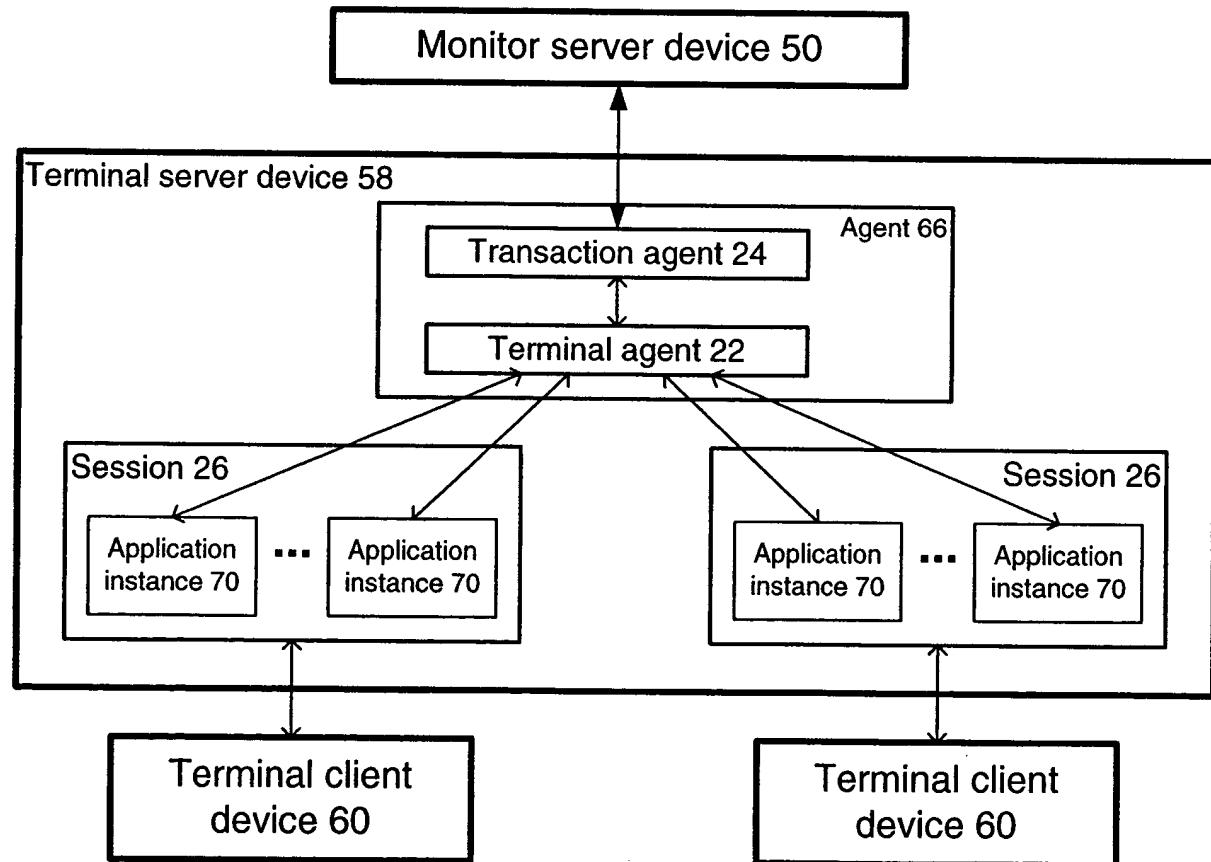
# FIG. 2C



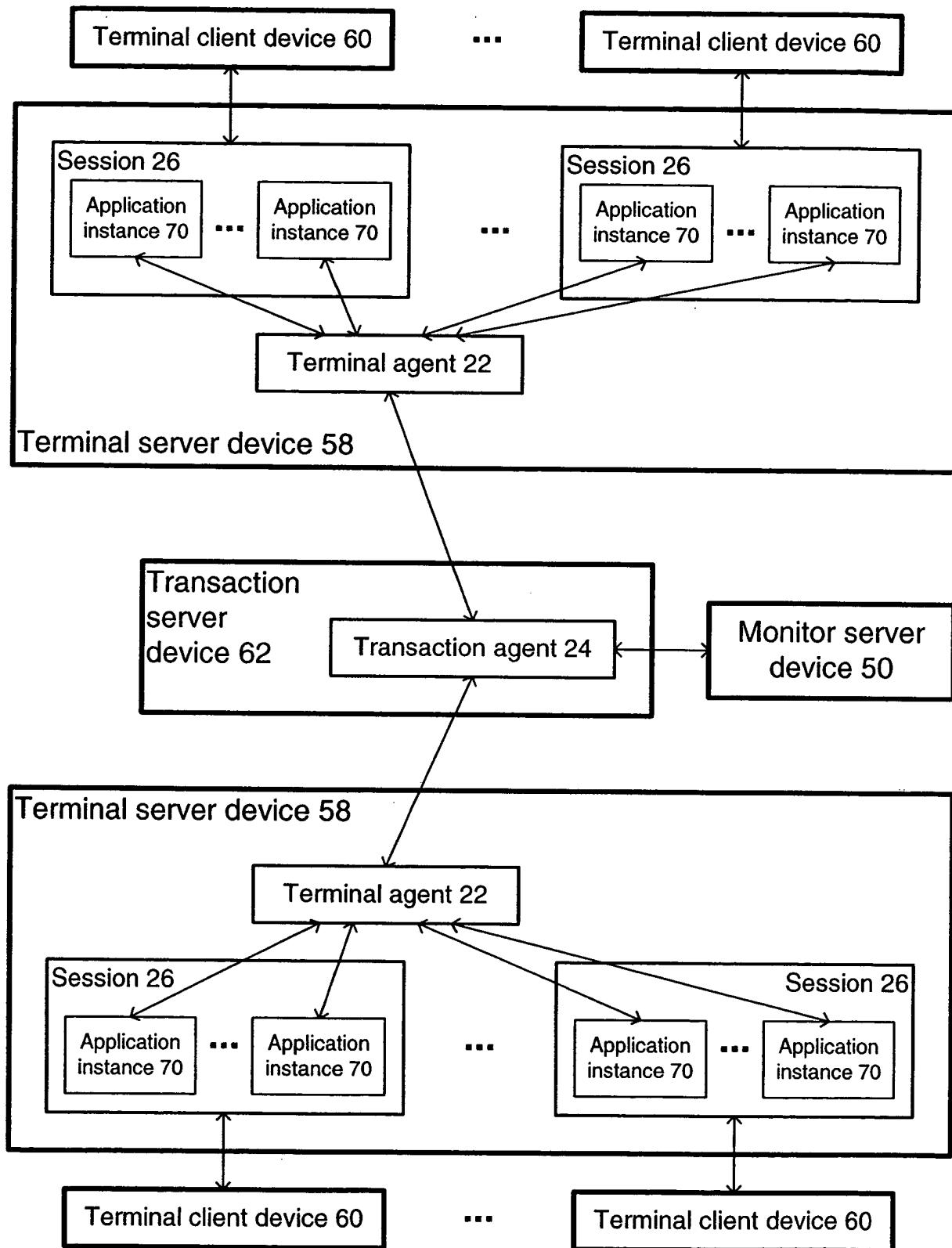
## FIG. 3A



## FIG. 3B

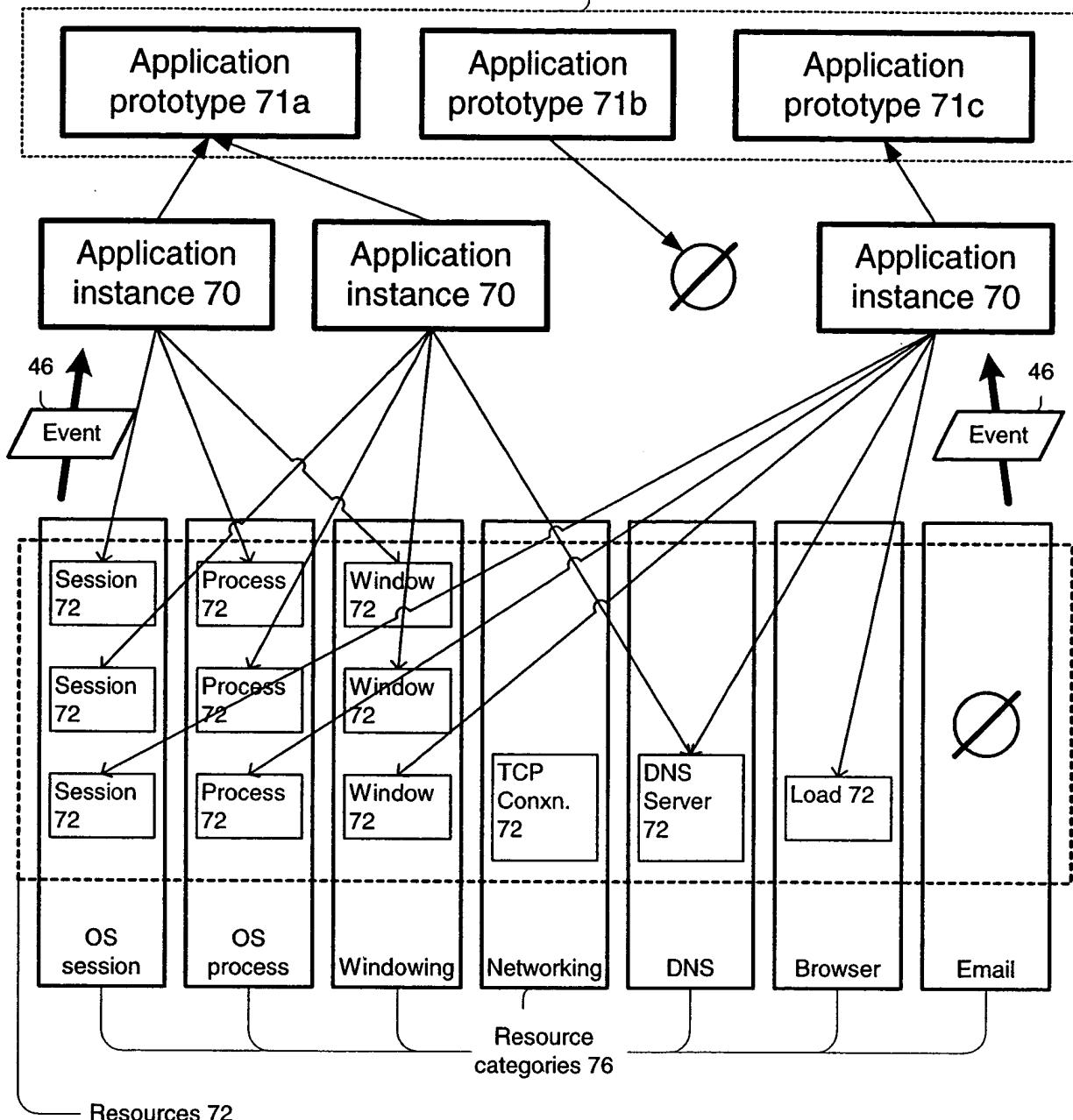


# FIG. 3C

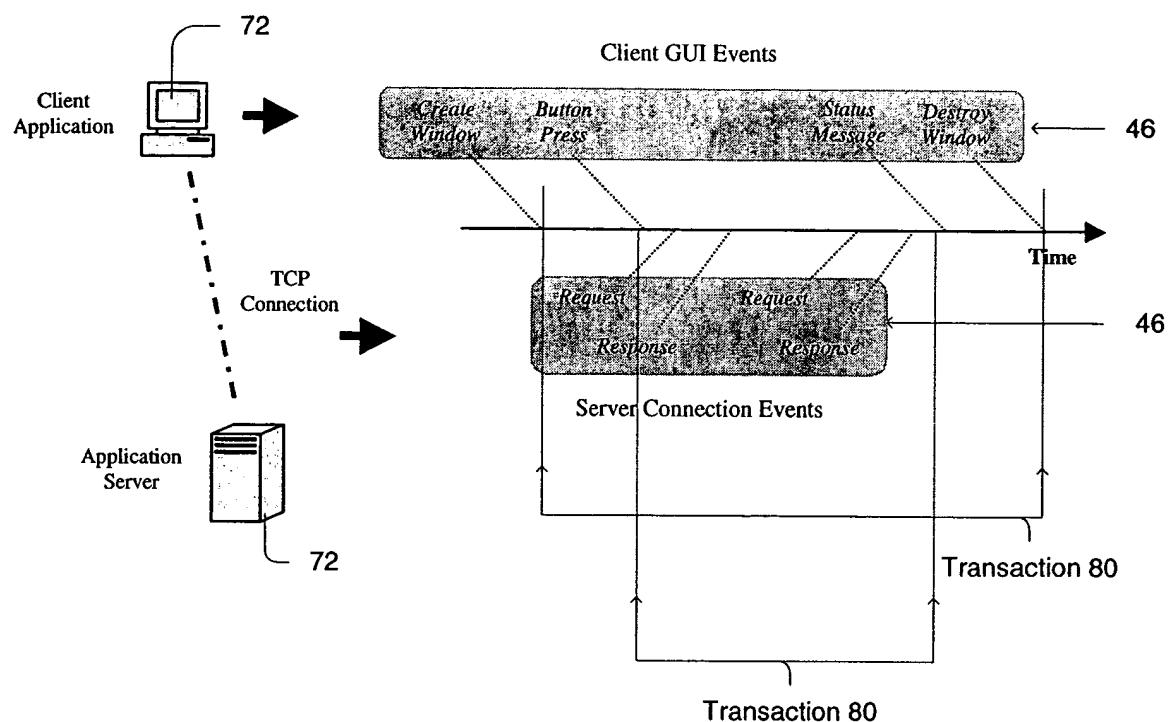


**FIG. 4**

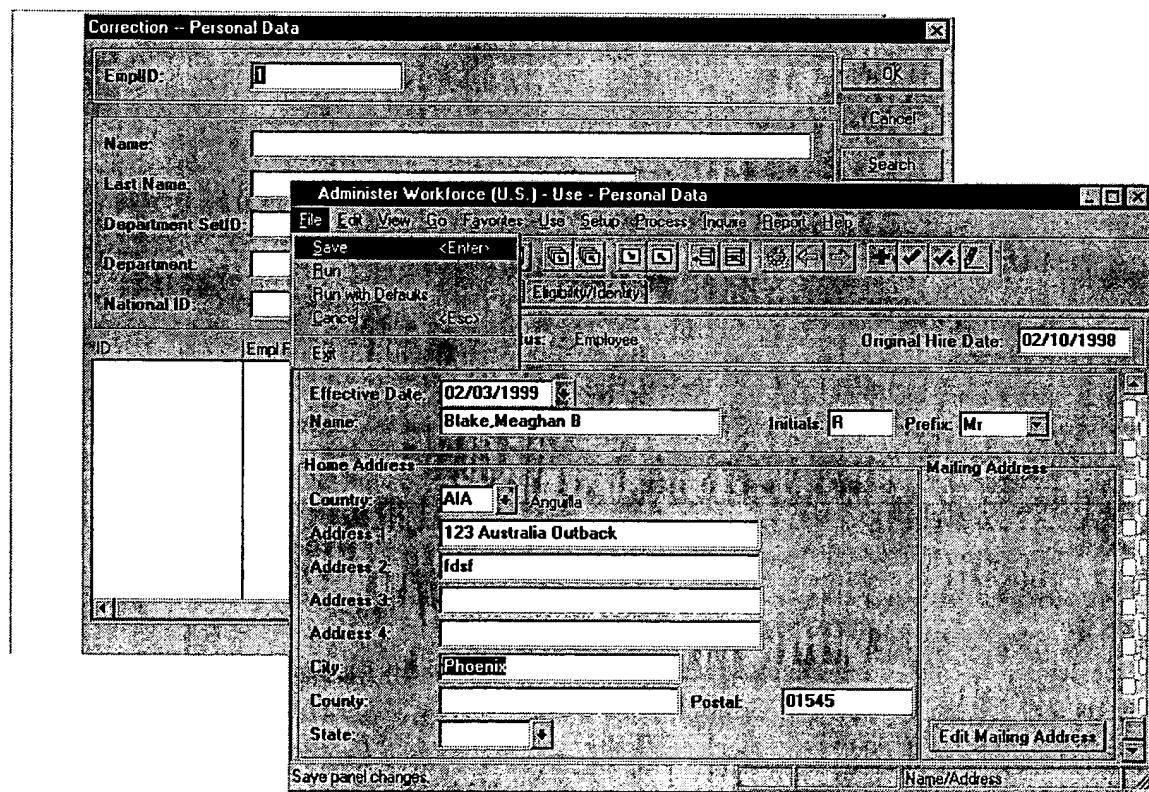
Application prototypes 71



## FIG. 5

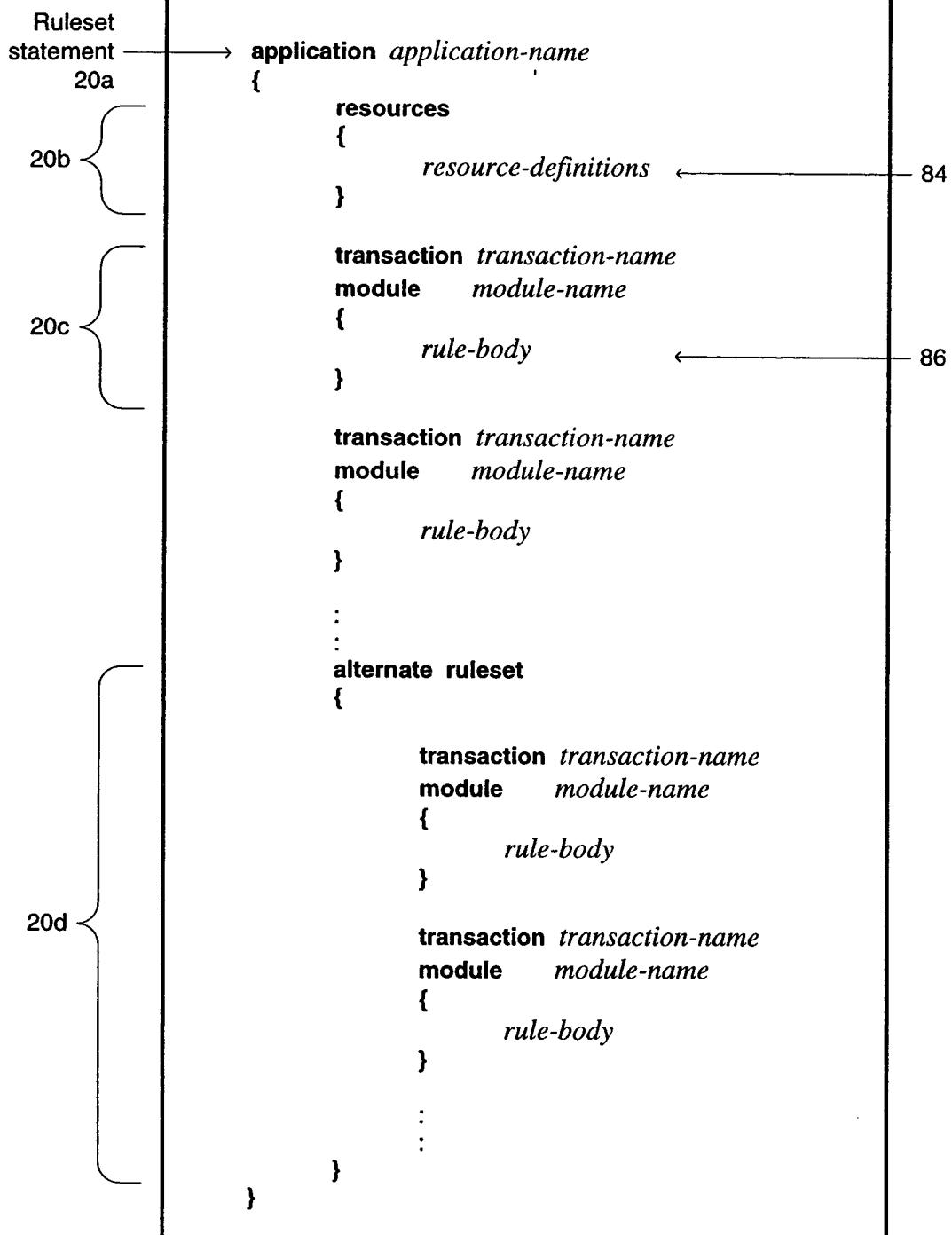


## FIG. 6

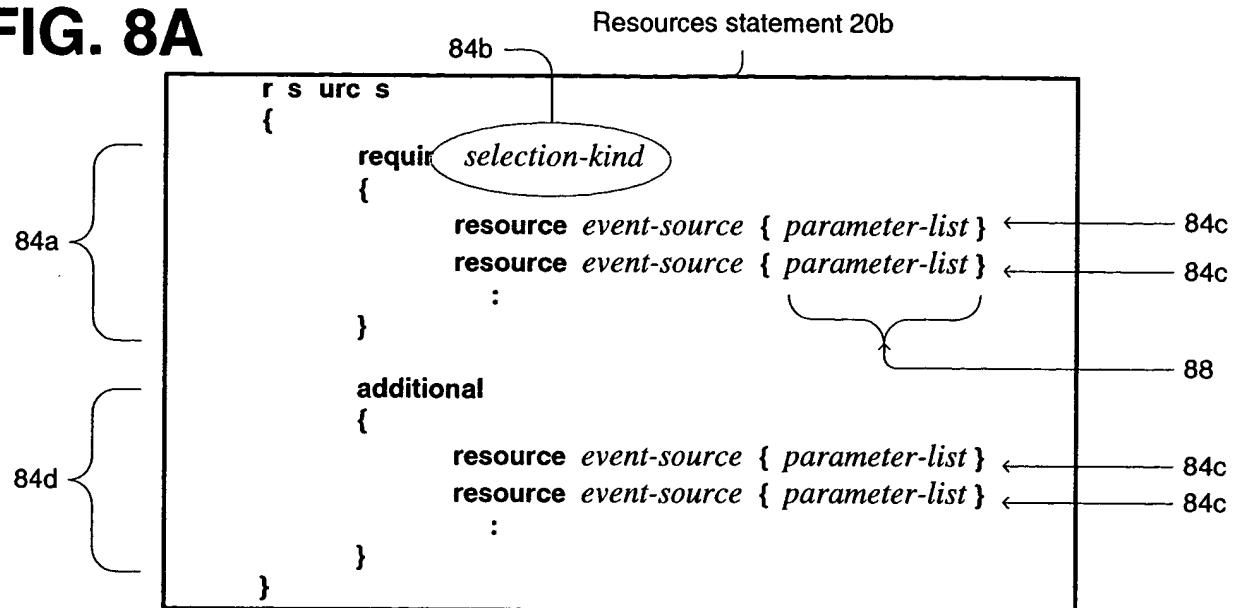


# FIG. 7

## Ruleset syntax 20



## FIG. 8A



## FIG. 8B

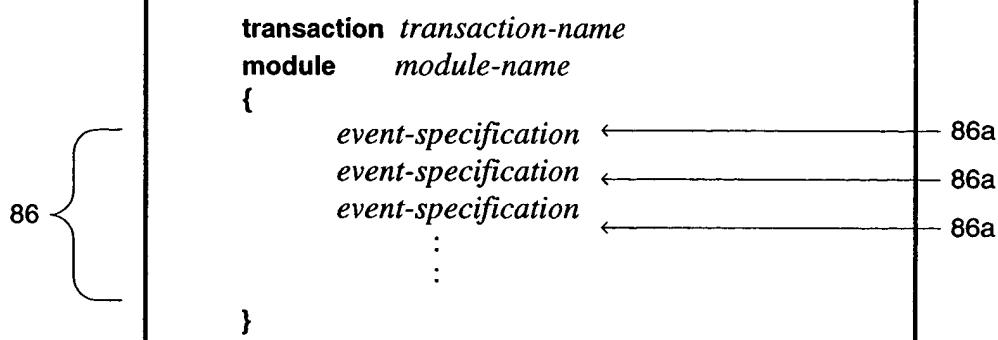
```
84b
resources
{
    require one
    {
        resource Process { ExecutableName="pstools" }
        resource Process { ExecutableName="pside" }
    }
    additional
    {
        resource Windows { }
        resource Connection { }
    }
}
```

## FIG. 8C

```
84b
resources
{
    require all
    {
        resource Process { ExecutableName="front" }
        resource Process { ExecutableName="sappui" }
    }
    additional
    {
        resource Windows { }
        resource Connection { }
    }
}
```

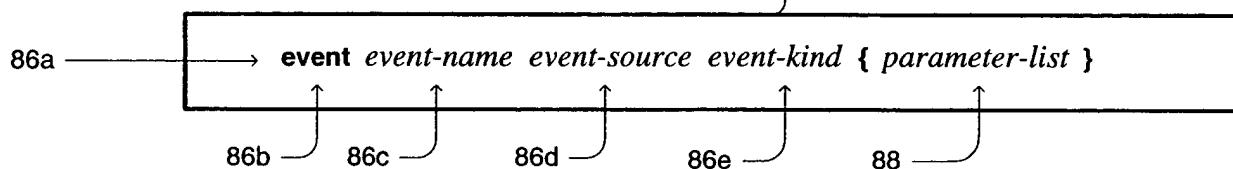
## FIG. 9A

Transaction statement 20c



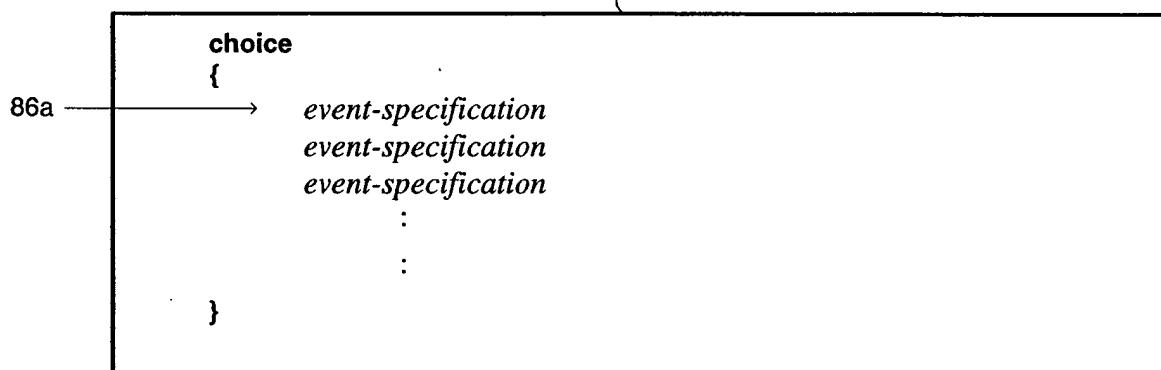
## FIG. 9B

Atomic event specification 86a



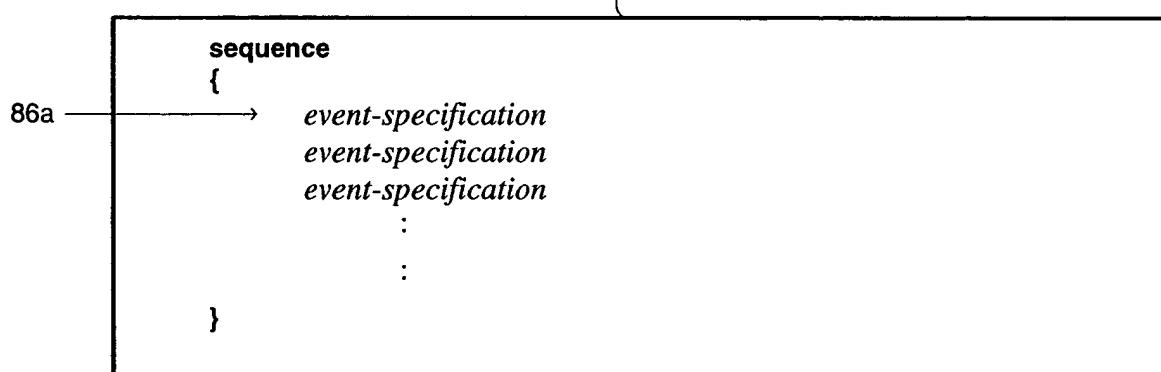
## FIG. 9C

Choice construct 86g

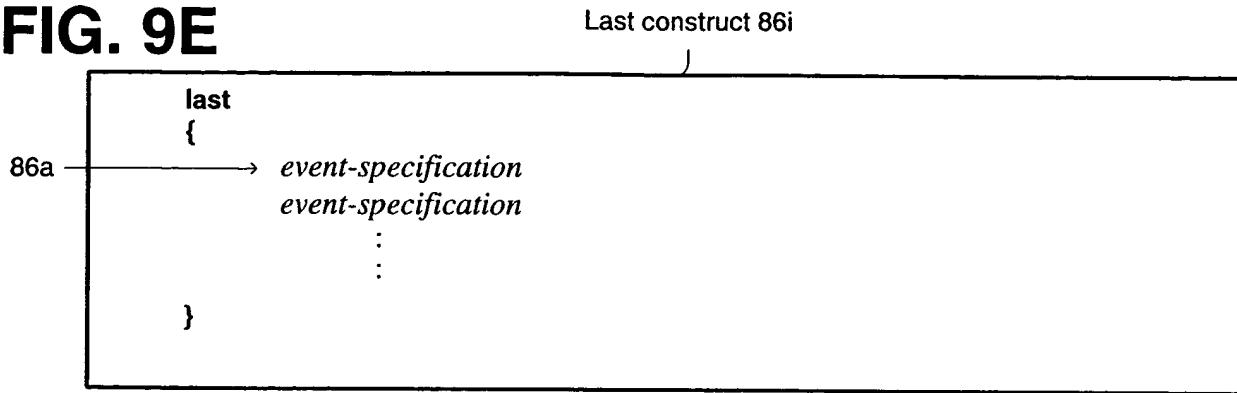


## FIG. 9D

Sequence construct 86h



## FIG. 9E



## FIG. 9F

```
transaction "UpdateEmployeeRecord"
module "PrimaryOperations"
{
    event "1-of-5" Windows SetFocus { Title="Correction - Personal Data" }
    event "2-of-5" Windows ButtonPress { Text="OK" }
    event "3-of-5" Windows SetFocus
    {
        Title="Administer Workforce (U.S.) - Use - Personal Data"
    }
    event "4-of-5" Windows MenuCommand { Text="File->Save" }
    event "5-of-5" Windows StatusMessage { Text="Record Saved" }
}
```

## FIG. 9G

```
transaction "AddNewEmployee"
module "PrimaryOperations"
{
    choice
    {
        sequence # Manually enter employee data
        {
            event "1-of-3(A)" Windows SetFocus { Title="Enter Employee Data" }
            event "2-of-3(A)" Windows MenuCommand { Text="Record->Save" }
            event "3-of-3(A)" Windows StatusMessage { Text="Record Saved" }
        }

        sequence # Import employee data from a file
        {
            event "1-of-3(B)" Windows SetFocus { Title="Import Employee Record" }
            event "2-of-3(B)" Windows MenuCommand { Text="Record->Save" }
            event "3-of-3(B)" Windows StatusMessage { Text="Record Imported" }
        }
    }
}
```

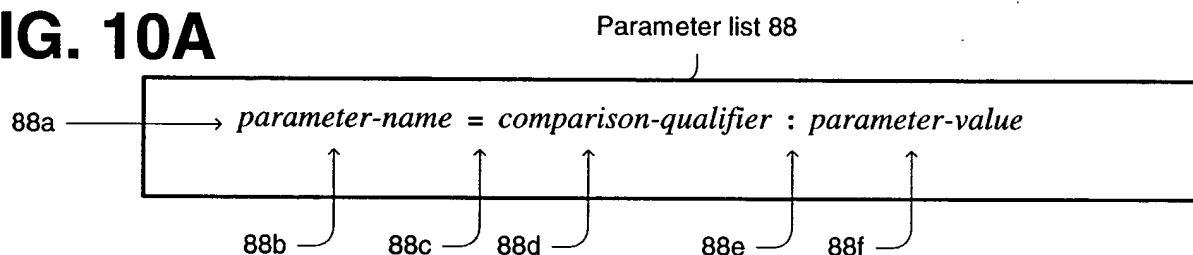
## FIG. 9H

```
transaction "ScheduleAppointment"
module    "Calendar"
{
    # Begin with the pressing of the "Schedule" button.
    event "1-of-2" Windows ButtonPress
    {
        ParentTitle="New Appointment" Text="Schedule"
    }

    # End with the last networking event to the database server.
    last
    {
        event "2(A)-of-2" Connection Request { }
        event "2(B)-of-2" Connection Response { }
    }
}
```

86i { } (brace grouping the last block)

## FIG. 10A



## FIG. 10B

```
event "1-of-7" Windows SetFocus { Title=contains:"Update Employee" }
```

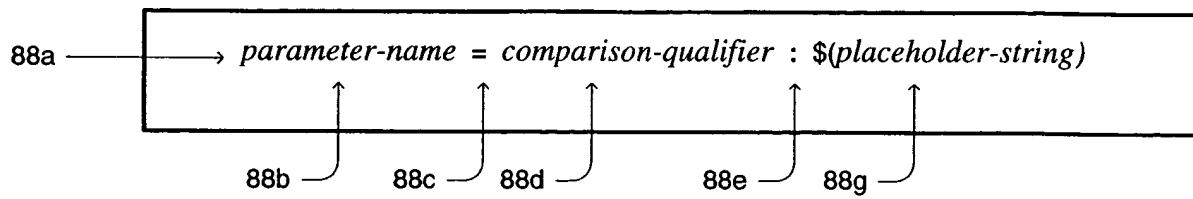
Parameter entry 88a

## FIG. 10C

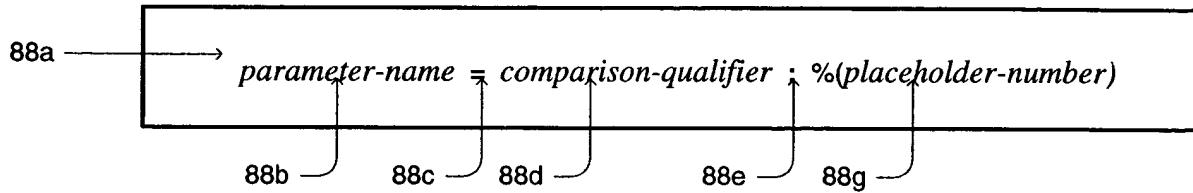
```
event "1-of-7" Windows SetFocus { Title=regexp:"Update Employee - .*" }
```

Parameter entry 88a

## FIG. 11A



## FIG. 11B



## FIG. 11C

Example ruleset 16

```
graph TD; 86a[86a] --> 88g[88g]; 88g --> 88a[88a];
```

```
application "Windows"
{
    resources
    {
        require one
        {
            resource Process { ExecutableName=$(Application Executable) }
        }
        additional
        {
            resource Windows { }
            resource Connection { }
        }
    }

    transaction "WindowTransition"
    module "AR"
    {
        choice
        {
            # Start with any window.
            event "Wnd1Title" Windows SetTitle { }
            event "Wnd1Focus" Windows SetFocus { }
        }

        choice
        {
            # Ends with any other window
            event "Wnd2Title" Windows SetTitle { }
            event "Wnd2Focus" Windows SetFocus { }
        }
    }
}
```

## FIG. 11D

```
resources
{
    require one
    {
        resource Process { ExecutableName=$(Application Executable) }
        resource Process { ExecutableName="CERNER" }
        resource Process { ExecutableName="CERNADV" }
    }
    additional
    {
        resource Windows { }
        resource Connection { }
    }
}
```

86a → 88g

## FIG. 11E

Example ruleset 16

```
application "Web"
{
    resources
    {
        require one
        {
            resource Web { }
        }
        additional
        {
            resource Connection { }
        }
    }

    transaction "BrowseAnywhere"
    module "AR"
    {
        # Begin page download.
        choice
        {
            event "BeginDownload" Web BeginLoad { URL=contains:$(URLs) }
        }

        # End page download.
        event "EndDownload" Web EndLoad { }
    }
}
```

86i { → 88g

## FIG. 11F

```
transaction "BrowseAnywhere"
module "AR"
{
    # Begin page download.
    choice
    {
        event "BeginDownload" Web BeginLoad { URL=contains:$(URLs) }
        event "BeginDownload" Web BeginLoad { URL=contains:"www.concord.com" }
        event "BeginDownload" Web BeginLoad { URL=contains:"www.irs.ustreas.gov" }
    }

    # End page download.
    event "EndDownload" Web EndLoad { }
}
```

## FIG. 11G

```
transaction "BrowseAnywhere"
module "AR"
{
    # Begin page download.
    event "BeginDownload" Web BeginLoad { URL=contains:$(URLs) }
    event "BeginDownload" Web BeginLoad { URL=contains:"www.concord.com" }
    event "BeginDownload" Web BeginLoad { URL=contains:"www.irs.ustreas.gov" }

    # End page download.
    event "EndDownload" Web EndLoad { }
}
```

## FIG. 11H

```
resources
{
    require one
    {
        resource Process { ExecutableName="CERNER" }
    }
    additional
    {
        resource Windows { }
        resource Connection { Hostname=$(Server Host) Port=%(Server Port) }
    }
}
```

# FIG. 12

Example ruleset 16

```
application "EmployeeManagement"
{
    resource s      # The application consists of a single executable.
    {
        require one
        {
            resource Process { ExecutableName="EmplMgmt" }
        }

        additional
        {
            resource Windows { }
            resource Connection { }
        }
    }

    #
    # User-level transactions. Only monitor new employee and employee
    # update operations.
    #

    transaction "NewEmployee"
    module    "UserOperations"
    {
        event "1-of-2" Windows MenuCommand { Text="New->Record..." }
        event "2-of-2" Windows StatusMessage
        {
            Text="New employee record created."
        }
    }

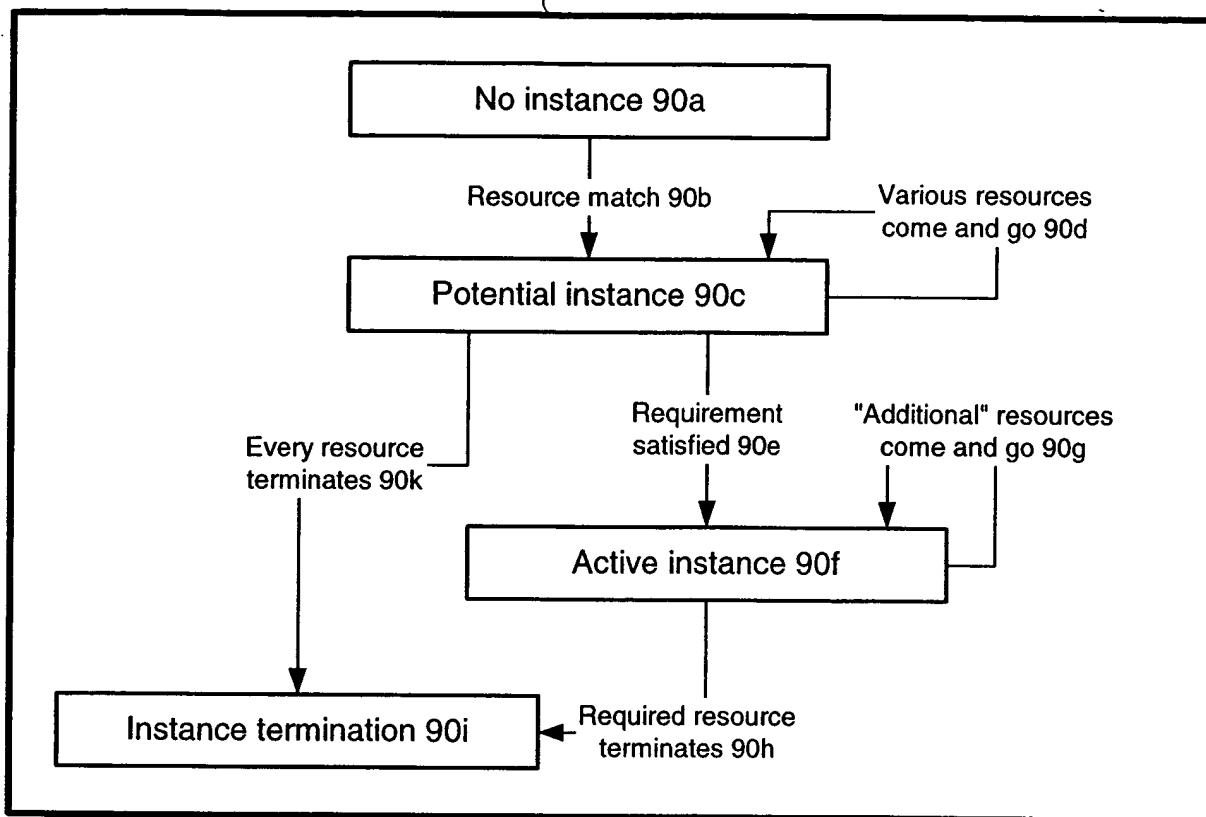
    transaction "EmployeeUpdate"
    module    "UserOperations"
    {
        event "1-of-2" Windows MenuCommand { Text="Edit->Save" }
        event "2-of-2" Windows StatusMessage { Text="Employee record updated." }
    }

    #
    # AR-level transactions. Monitor every window transition.
    #

    alternate ruleset
    {
        transaction "ARWindowTransition"
        module    "AROperations"
        {
            event "1-of-2" Windows SetFocus { }
            event "2-of-2" Windows SetFocus { }
        }
    }
}
```

## FIG. 13

Application lifecycle 90



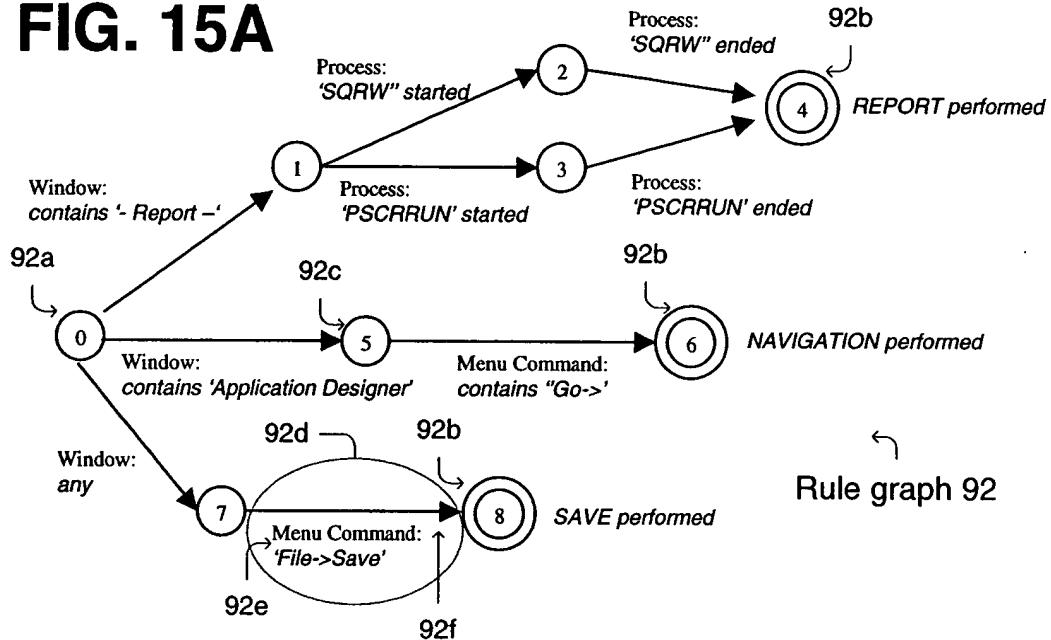
## FIG. 14

Ruleset engine 36

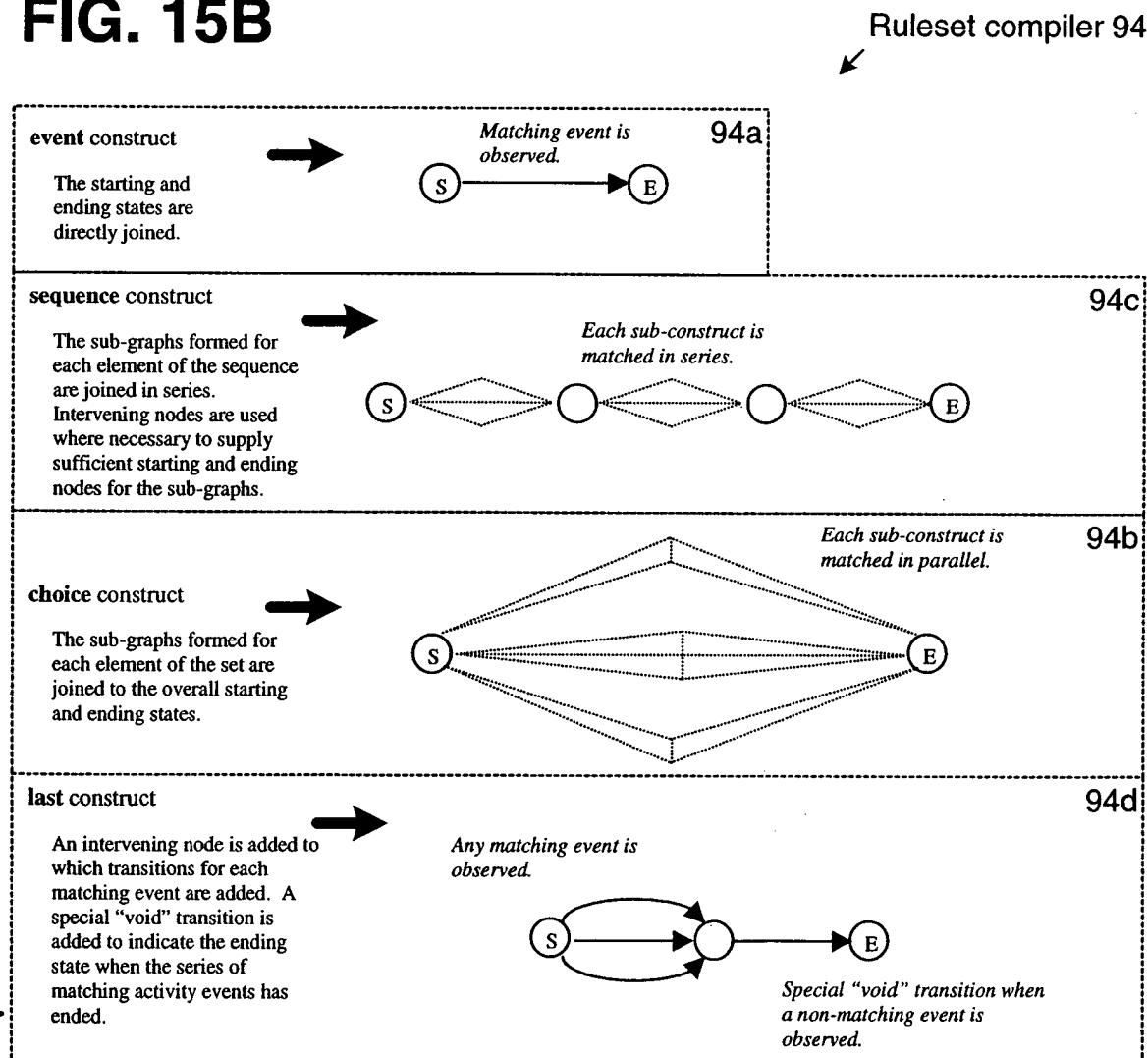
Ruleset compiler 94

Transaction recognition process 98

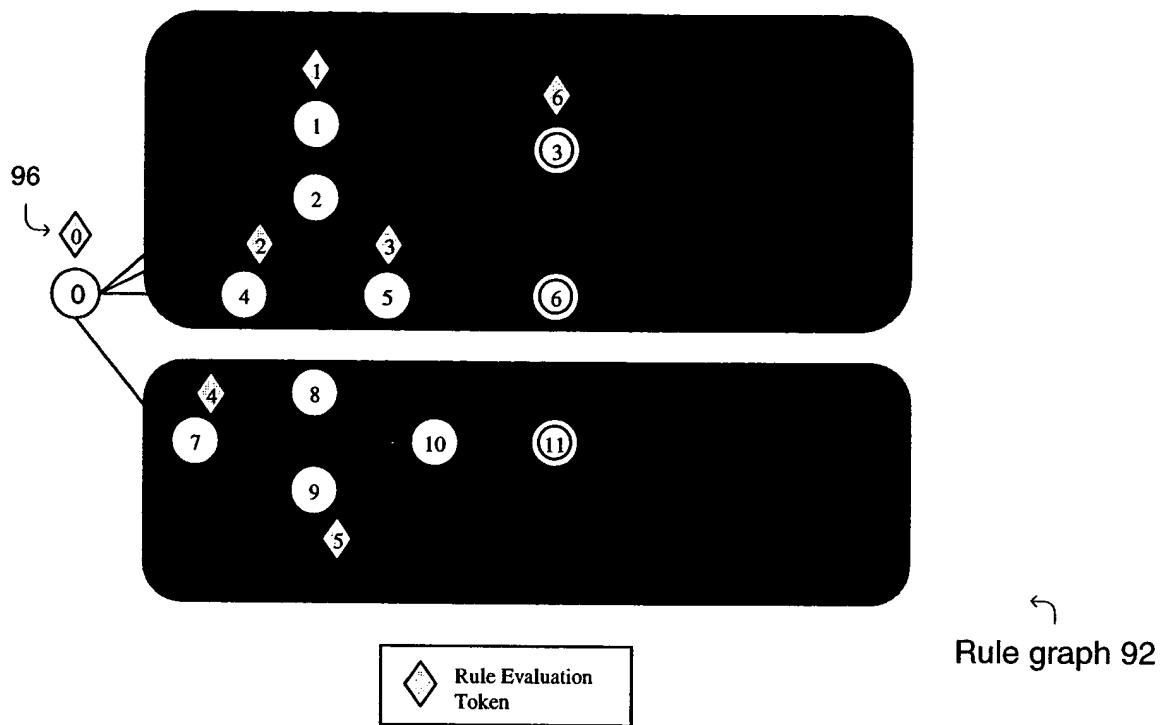
# FIG. 15A



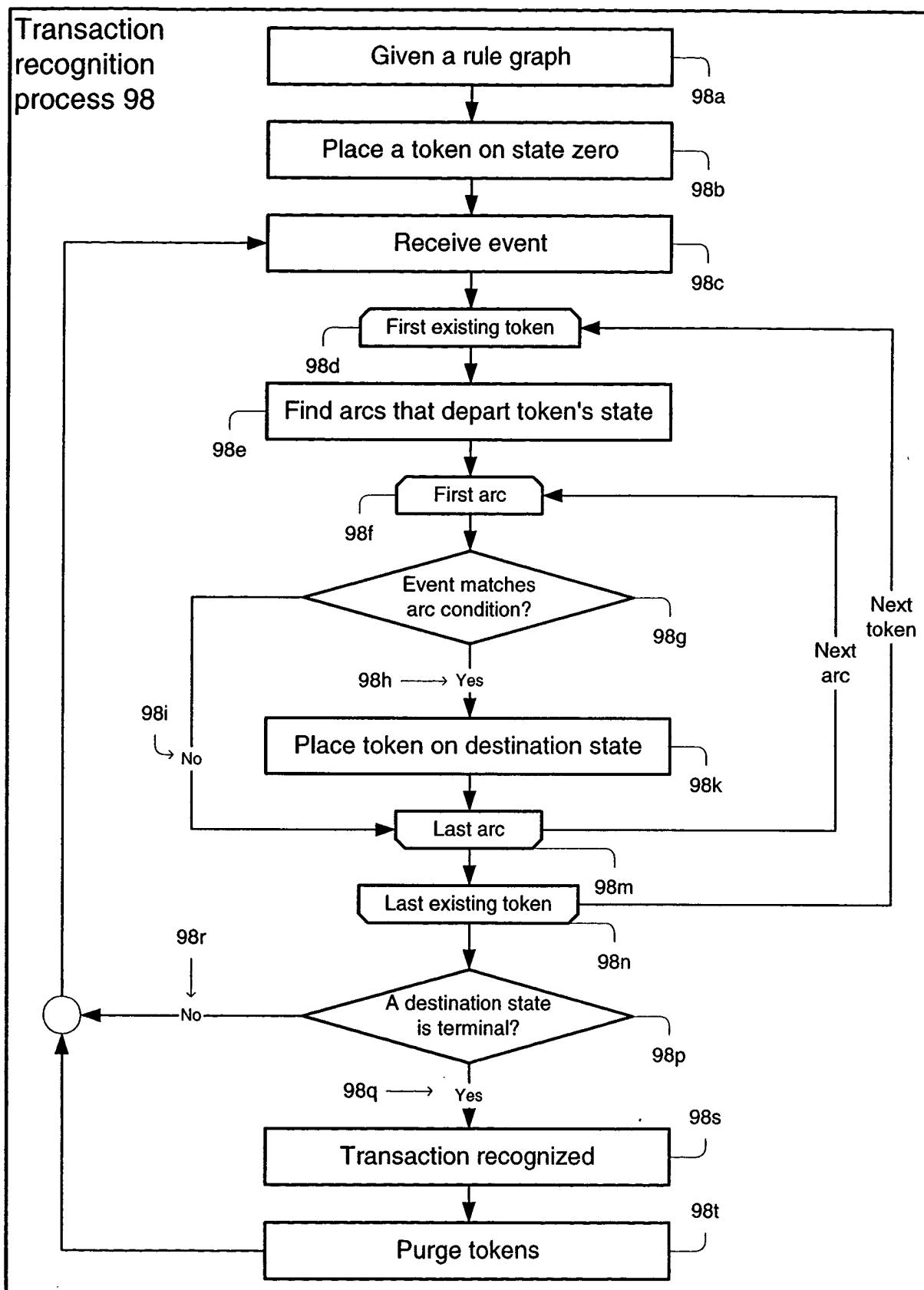
# FIG. 15B



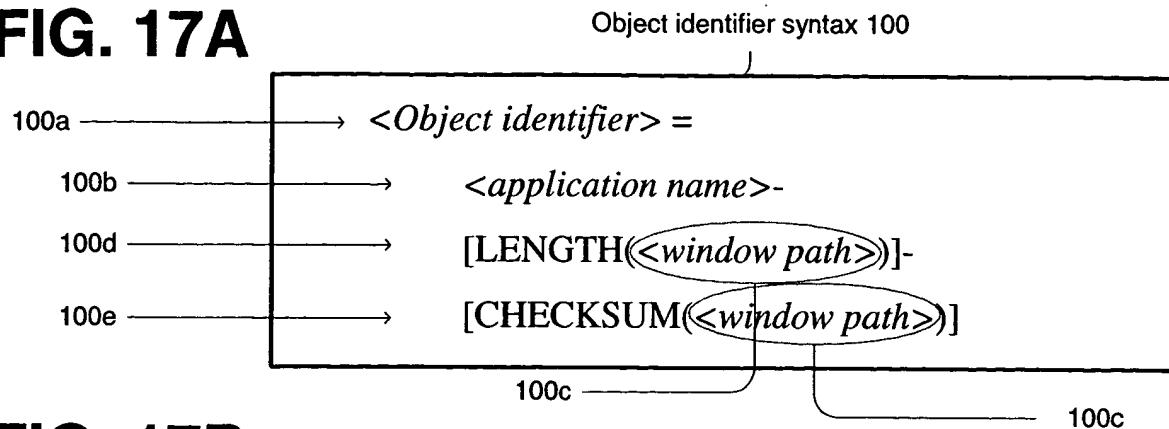
# FIG. 15C



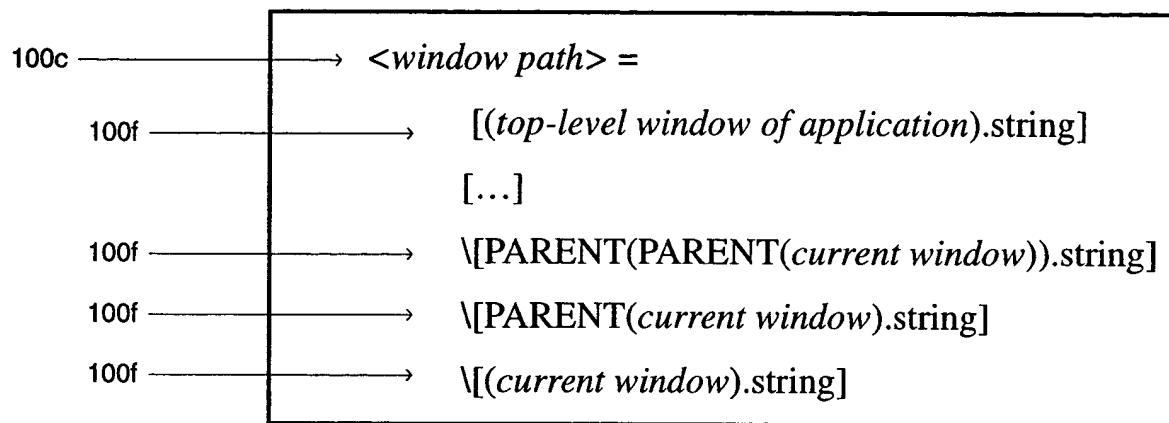
# FIG. 16



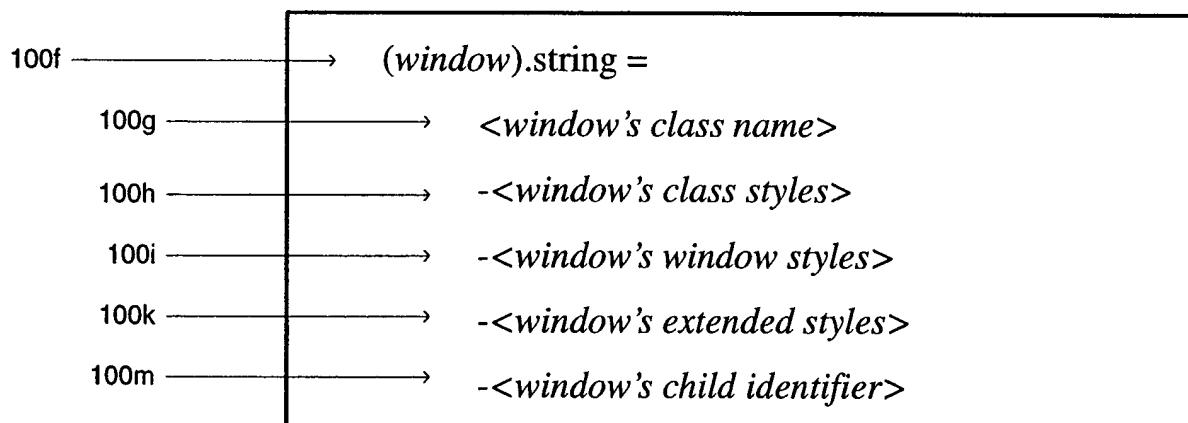
# FIG. 17A



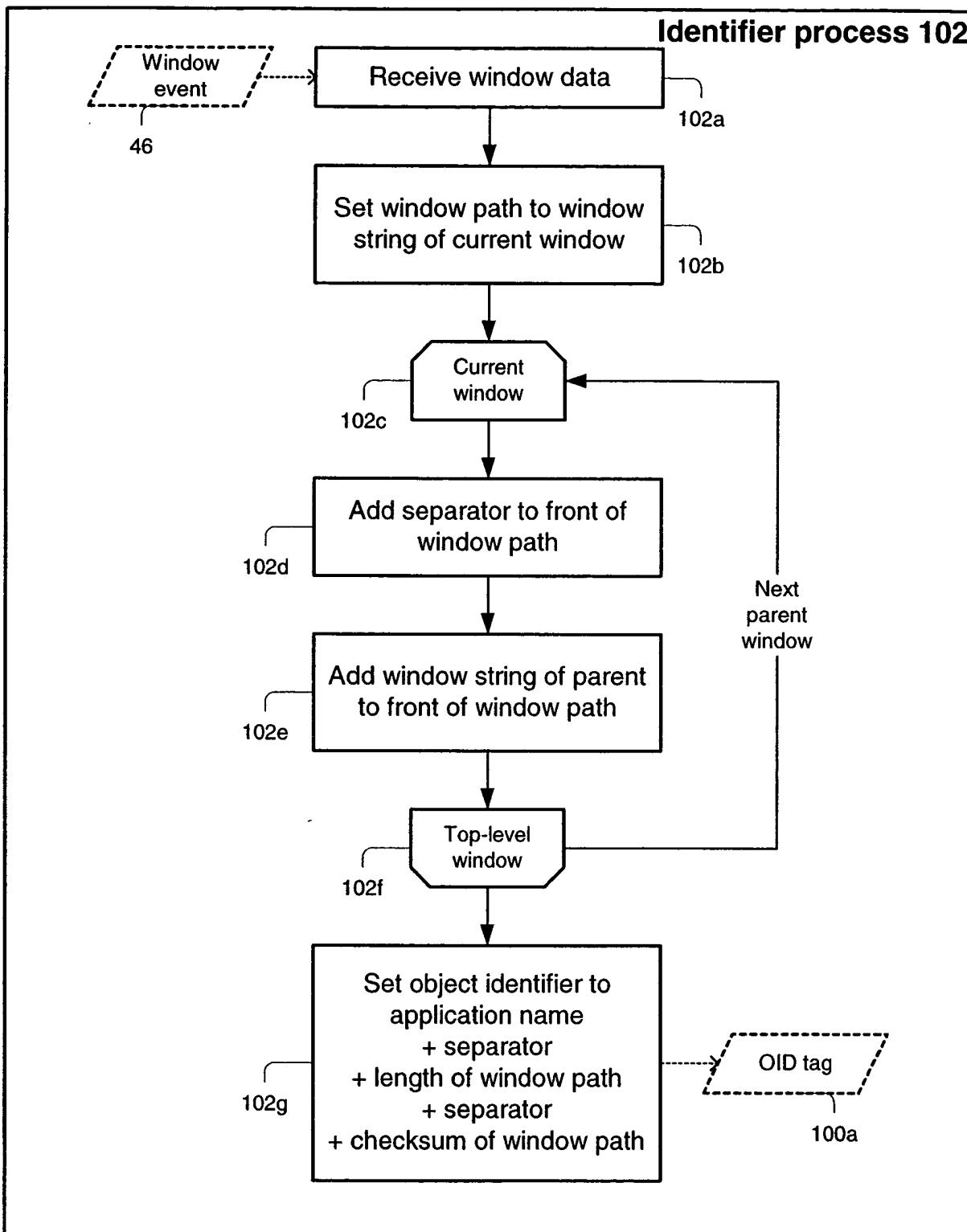
# FIG. 17B



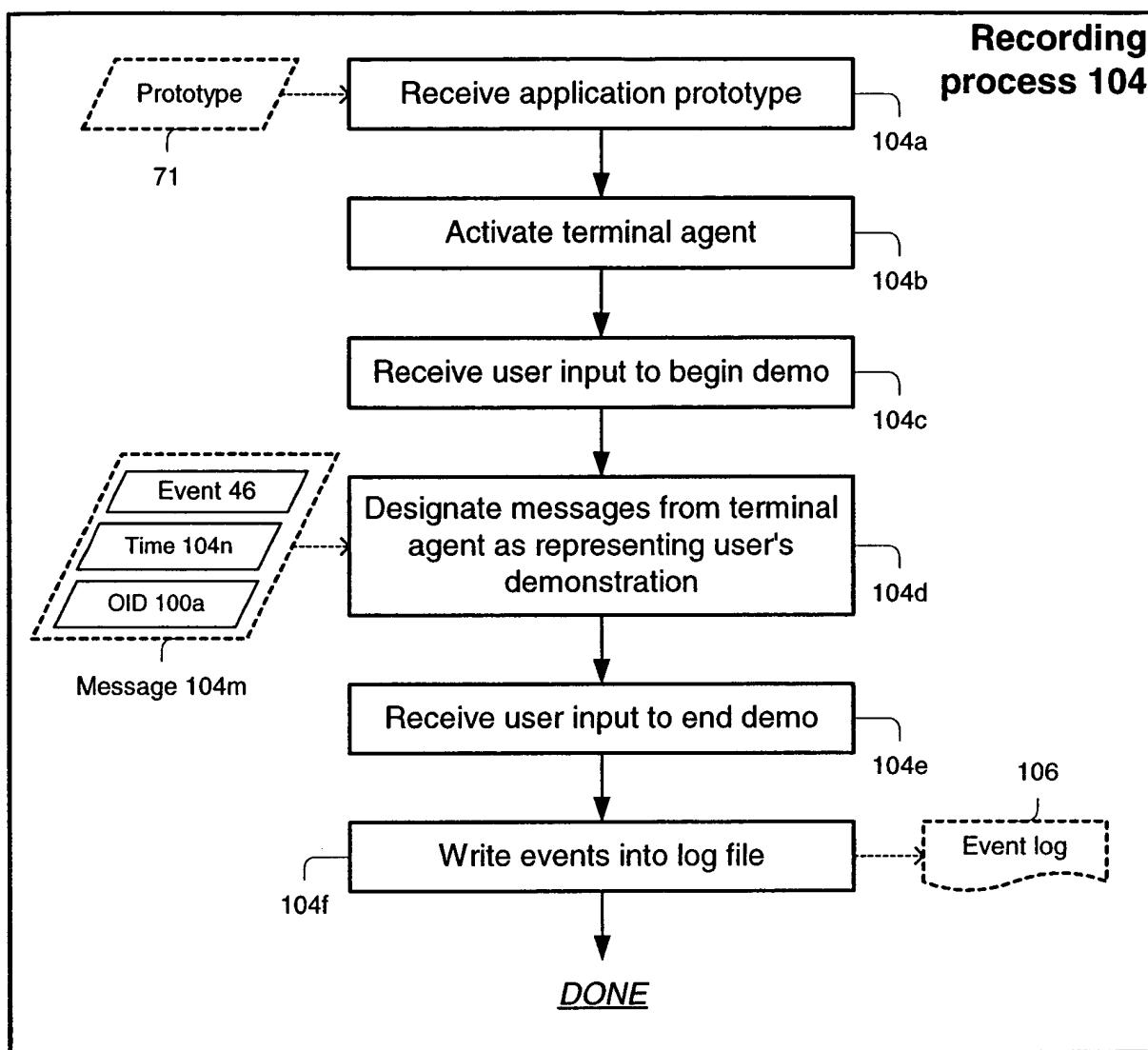
# FIG. 17C



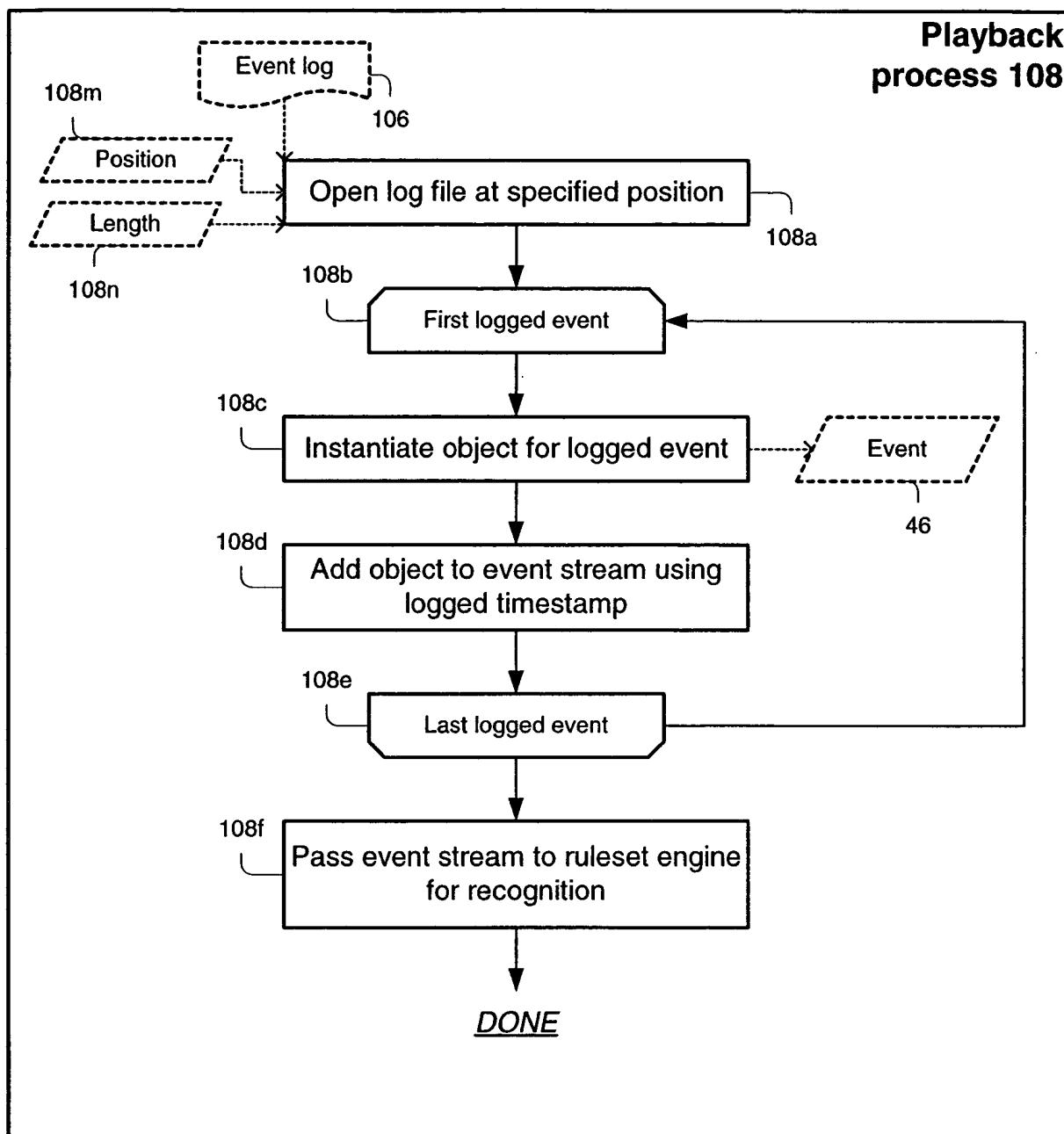
# FIG. 18



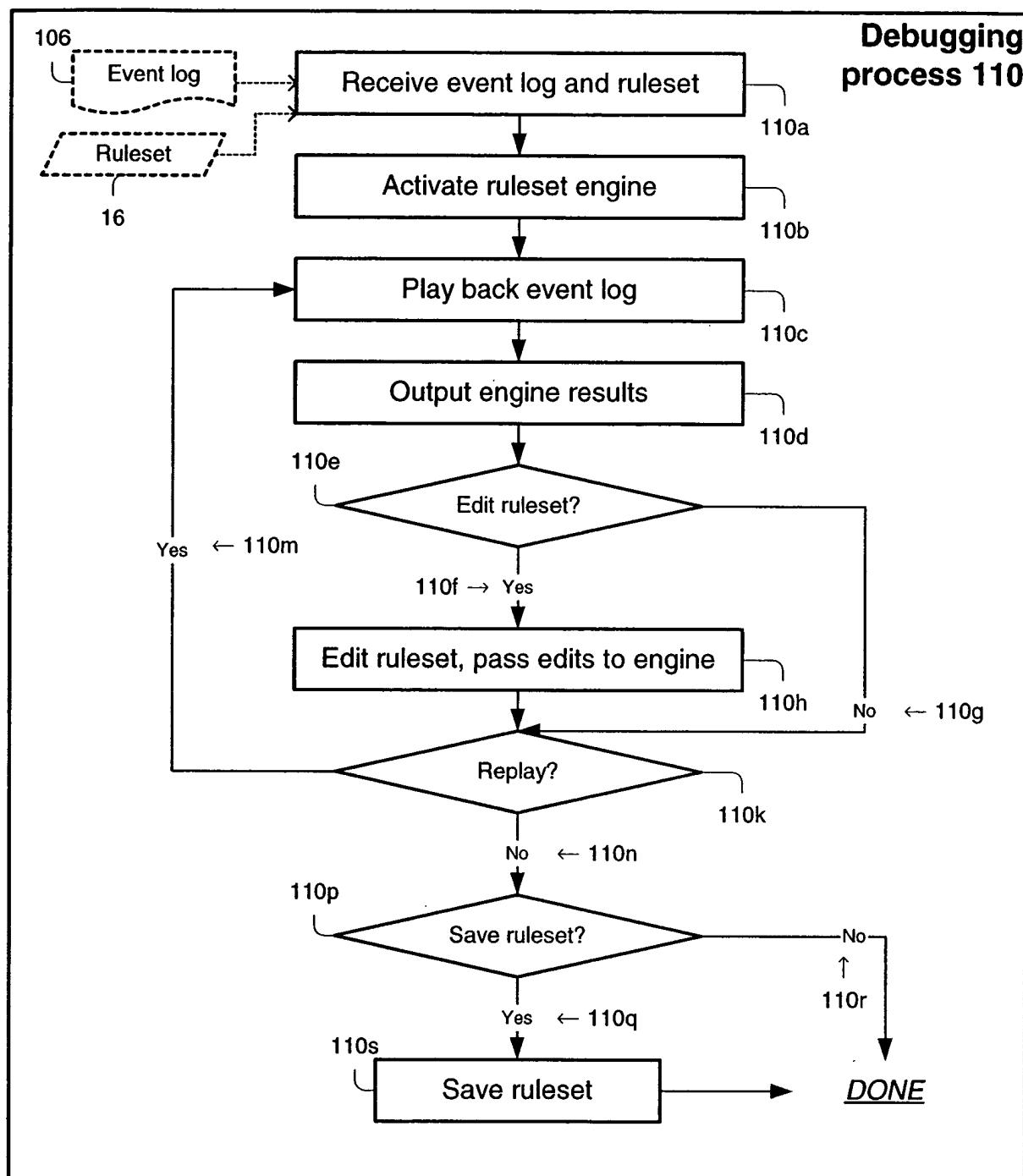
# FIG. 19A



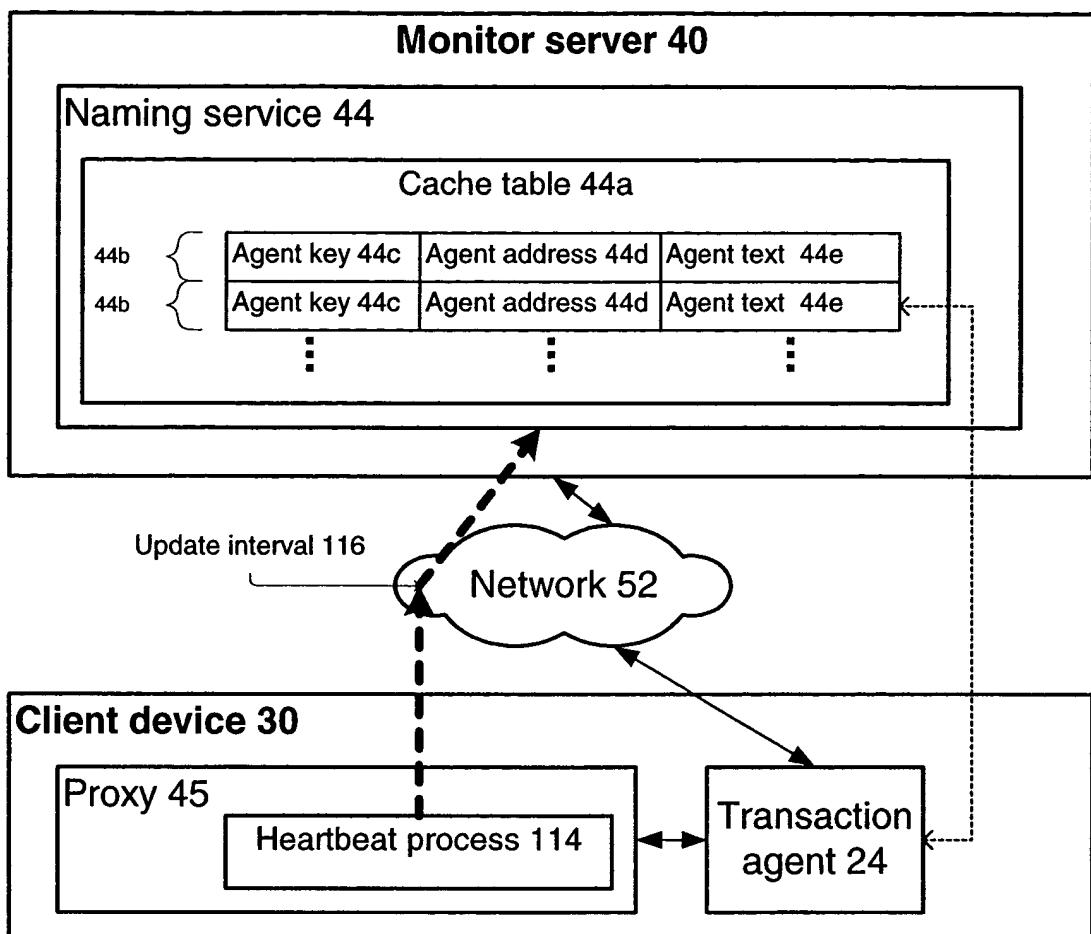
# FIG. 19B



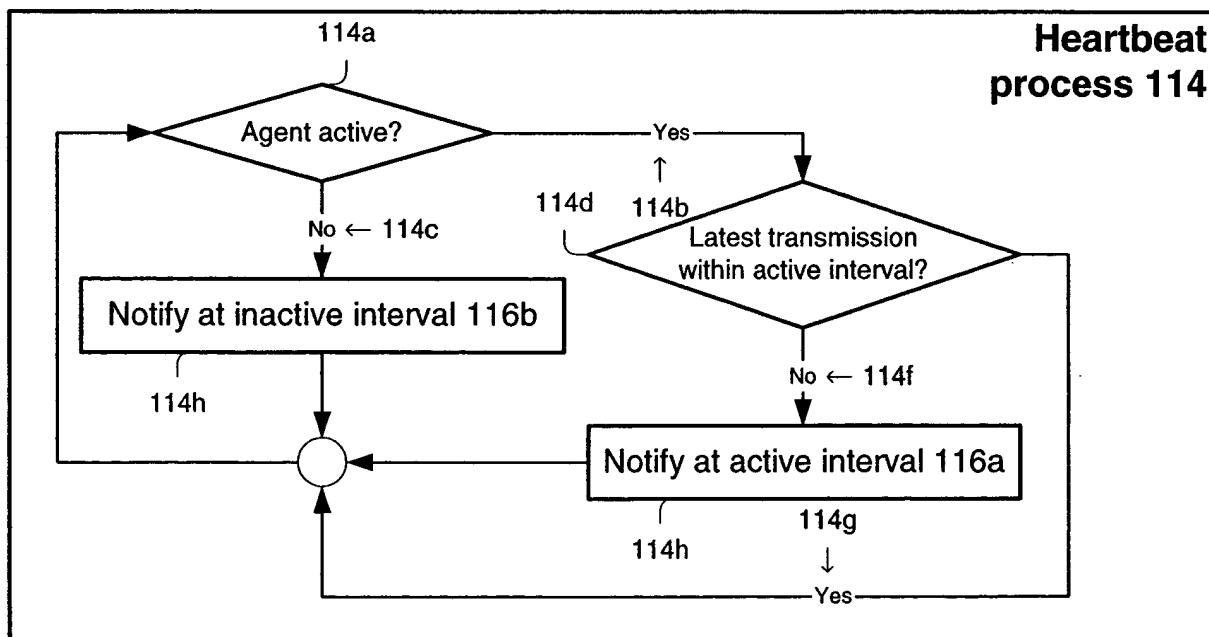
# FIG. 19C



**FIG. 20A**



**FIG. 20B**



# FIG. 21A

## Pathset aggregation 42

## Pathset aggregation proc ss 120

### Pathset aggregation data 118

Network elements 118a

Client 118b

Server 118c

Network  
infrastructure 118d

Client set 118e

Pathset 118g

Responsiveness  
samples 118h

Clientset ref. 118k

Module ref. 118m

Module 118f

# FIG. 21B

## Pathset aggregation process 120

Transaction  
event

Receive transaction statistics

38

## Pathset aggregation proc ss 120

120b

Match to pathset by client set  
and module

120c

Add statistics to responsiveness  
sample

DONE